IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re U.S. Patent Application of:	)	**Group Art Unit: 2137**
	)	
Harald VATER *et al.*	)	**Examiner: Z. Davis**
	)	
Serial Number: 09/700,656	)	***Attorney Docket*: VATE3001/BEU**
	)	
Filed: February 14, 2001	)	**Confirmation No.: 2137**

For: Access-Controlled Data Storage Medium

APPELLANT'S BRIEF UNDER 37 C.F.R. §41.37

Sir:

This paper is a second Appeal Brief in furtherance of the Notice of Appeal filed concurrently herewith. Notice of Appeal and Appeal Brief fees have been previously submitted in connection with an appeal that was terminated upon re-opening of prosecution by the Examiner.

This Brief contains these items under the following headings and in the order set forth below:

I. Real Party In Interest

The real party in interest is Giesecke & Devrient, GmbH, of Munich, Germany.

II. Related Appeals And Interferences

There are no related appeals or interferences.

III. Status of Claims

The status of the claims in this application is:

A. Status of all the claims

1. Claims canceled: 1-25, 34-41, and 43
2. Claims withdrawn from consideration: None
3. Claims pending: 26-33 and 42
4. Claims allowed: None
5. Claims objected to: None
6. Claims rejected: 26-33 and 42

B. Claims on Appeal:

The claims on appeal are: 26-33 and 42

IV. Status of Amendments

No amendments have been submitted subsequent to the rejection mailed January 20, 2011.

V. Summary of Claimed Subject Matter

The claimed subject matter is a method for protecting secret input data in which falsification of the input data (by combination with auxiliary data (Z)) is compensated for by combining output data that results from execution of operations *f* with an auxiliary function value *f*(Z) that has been **retrieved from a memory of the semiconductor chip** after being "***previously determined*** *by execution of the one or more operations a (f) with the auxiliary data (Z) as input data in safe surroundings and stored along with the auxiliary data (Z).* In other words, the claimed invention

is to falsify input data and yet achieve the same result as if the input data had not been falsified by using a **stored** auxiliary function value that was previously generated in safe surroundings. This feature of the invention, recited as the third and fifth steps of **independent claim 26**, is described in **lines 17-19 on page 3 and lines 6-9 on page 7** of the original specification. The first, second, and fourth steps recited in claim 26 are the falsification of input data, execution of operations *f* on the semiconductor chip, and combination of output data with the auxiliary function value, as described for example in **lines 3-17 on page 3 and lines 2-13 on page 7** of the original specification, and illustrated as **steps 9 and 10 of Fig. 3**.

It is of course necessary, whenever falsifying input data, to apply some sort of auxiliary function in order to obtain an intended result, *i.e.*, one that is not affected by the input data falsification. For example, the Kocher publication cited by the Examiner and discussed below (U.S. Patent Publication No. 2002012478), discloses such input data falsification using auxiliary data and auxiliary function values. However, a problem with falsification of input data using auxiliary data and auxiliary function values is that it might be possible for an attacker to determine the auxiliary data and auxiliary function values as they are generated. **This problem is solved in the claimed invention by using auxiliary data (*Z*) and auxiliary function values *f*(*Z*) <u>that have been previously calculated</u> in safe surroundings, <u>stored</u> for use in connection with the input data falsification, and <u>retrieved from a memory</u> on the chip for combination with the output data upon execution of operations (*f*)**.

The significance of storing the auxiliary function *f*(Z) in safe surroundings is explained at the end of the second paragraph on page 3 of the present application:

> . . .It is important in this context for the random number and the function value [*i.e.*, the auxiliary data and the auxiliary function value] to be previously determined and stored in safe surroundings so that the calculation of the function value from the random number cannot be intercepted . . .."

Furthermore, as explained in line 7 on page 7 of the original specification, this previous determination could be made during production of the card or, more precisely (as explained in the last line of the third paragraph on page 8), during the personalization phase of card production.

As indicated above, **claim 26** positively recites five steps:

- falsifying the input data by combination with auxiliary data (Z) before execution of the one or more operations (f) on the semiconductor chip,
- executing said one or more operations (f) on the semiconductor chip,
- retrieving an auxiliary function value ($f(Z)$) from said memory of said semiconductor chip of the data carrier,
- combining the output data determined by said executing of the one or more operations (f) with said auxiliary function value ($f(Z)$) in order to compensate for the falsification of the input data, and
- previously determining the auxiliary function value ($f(Z)$) by execution of the one or more operations (f) with the auxiliary data (Z) as input data in safe surroundings, the previously-determined auxiliary function value being stored along with the auxiliary data (Z) in the memory of the semiconductor chip of the data carrier.

The first, second, and fourth steps, described in **lines 3-17 on page 3 and lines 2-13 on page 7**, are also disclosed in the prior art, but the ***combination*** of the first, second, and fourth steps with the third and fifth steps, described in lines **17-19 on page 3 and lines 6-9 on page 7**, is not disclosed in the prior art (although it is alleged by the Examiner to be obvious). In particular, the prior art does not disclose that the auxiliary function value used to compensate for the falsification of input data is ***previously*** determined by execution of the one or more operations with the auxiliary data as input data ***in safe surroundings***, **stored** in a memory on the chip, and **retrieved** from the memory, as claimed. The five steps listed above constitute all of the positive limitations in claim 26.

Dependent **claim 27** recites the feature in which falsification of the input is performed before execution of a nonlinear operation g. This feature is described in **lines 11-14 on page 3**, **lines 29-31**

on page 6, **lines 21-23 on page 7**, and especially **lines 1-3 on page 8** of the original specification, and simply refers to the fact that compensation may not be possible after application of a nonlinear rather than linear function (*f* being a linear function).

Dependent **claim 28** recites the feature of varying the auxiliary data to make discovery more difficult, as described in **lines 6-9 on page 9** of the original specification, while **claim 29** recites the feature of generating auxiliary data and functions by combining other auxiliary data and functions, as described in **lines 9-14 on page 9**.

Claim 30 depends from claim 29 recites the feature in which the auxiliary values and functions are selected randomly as described in **lines 6-14 on page 9**.

Dependent **claim 31** recites the feature described in **lines 9-14 on page 9** in which only some of the auxiliary values and functions are stored and others are generated from the previously determined values and functions without having to apply operation *f* to the auxiliary values.

Finally, with respect to the dependent claims, **claims 31 and 32** respectively recite the features, described for example in **lines 3-17-19 on page 3 and lines 6-15 on page 9**, that the auxiliary data are random numbers and the output data are EXORed with the auxiliary function, while **claim 42** recites the feature in which the operations to be protected are key permutations or permutations of other secret data as described in **lines 15-26 on page 8**.

Since the dependent claims all depend from claim 26 and therefore incorporate, by virtue of their dependency, the limitation of claim 26 that the auxiliary function is previously determined in safe surroundings, stored, and then retrieved for combination with output data, all of the claims involved in this appeal include this limitation..

VI. Grounds of Rejection to be Reviewed on Appeal

The rejection to be reviewed on appeal is a rejection of the subject matter of claims 26-33 and 42 as obvious under 35 USC §103(a) in view of U.S. Patent Publication No. 2001/0053220 (the Kocher publication) and U.S. Patent No. 5,655,023 (Cordery).

VII. Argument

Reversal of the rejection under 35 USC §103(a) is respectfully requested on the grounds that the Kocher publication and the Cordery patent, whether considered individually or in any reasonable combination, fail to disclose or suggest the claimed combination of:

- falsifying the input data by combination with auxiliary data (Z) before execution of the one or more operations (f) on the semiconductor chip,
- executing said one or more operations (f) on the semiconductor chip,
- retrieving an auxiliary function value ($f(Z)$) from said memory of said semiconductor chip of the data carrier,
- combining the output data determined by said executing of the one or more operations (f) with said auxiliary function value ($f(Z)$) in order to compensate for the falsification of the input data, and
- previously determining the auxiliary function value ($f(Z)$) by execution of the one or more operations (f) with the auxiliary data (Z) as input data in safe surroundings, the previously-determined auxiliary function value being stored along with the auxiliary data (Z) in the memory of the semiconductor chip of the data carrier.

as recited in claim 26. Claims 27-33 and 42 stand or fall with claim 26.

The Kocher publication discloses computation of an auxiliary function value and combination with output data, as claimed, but the auxiliary function value is not calculated in **safe surroundings** and **stored** in a memory on the chip, as claimed. To the contrary, it is a main objective of Kocher to perform operations in an **insecure environment**, while protecting the input

data by disguising both the input data and the operations performed on the input data. This is also true of the claimed invention. If the operations performed by Kocher were performed in a secure environment, then there would be no need to disguise the input data or the operations. Where Kocher differs from the claimed invention is that the functions used to disguise the operations in the insecure environment are calculated at the time the operations are performed, whereas the claimed invention pre-computes those functions. This does not mean that Kocher's method is insecure–the operations are still disguised while they are being performed and the input data is still protected. Based on the disclosure of Kocher, one of ordinary skill in the art would not have realized that there is any vulnerability at all. Instead, the ordinary artisan would understand from Kocher that auxiliary function values are computed and used to disguise operations, which in turn protects secret data used in the operations. It is only from Appellant's disclosure that one discovers that it might be possible to analyze emission patterns from the disguised operations to discover how the auxiliary function values are calculated, and to use that information to discover the nature of the operations being disguised.

In other words, both the claimed invention and Kocher seek to protect data used in operations that are performed in an insecure environment, by disguising both the input data and the operations being performed. The difference lies solely in how the auxiliary function values used to disguise the operations are calculated. In Kocher, the auxiliary function values are specifically calculated as part of the disguising algorithm, while the disguising algorithm of the claimed invention uses pre-computed function values. It is not obvious from Kocher that pre-computed function values are desirable, or even how they would be pre-computed.

Cordery, on the other hand, does not seek to protect input data by disguising operations being performed on the input data, or even disguising the input data. Cordery is not at all concerned with the problem of power analysis of operations performed on a chip, which could lead to revealing the operations being performed on the chip in order to reveal the secret data used in the operations. Instead, Cordery takes the approach that the data used in the operations being performed (printing

postage meter labels) may be protected by storing the input data on a removable storage device or token. As explained in col. 4, lines 54-63 of the Cordery patent, the purpose of Cordery's removable or portable tokens is to avoid storing secret keys, or requiring that secret keys be used during postage operations:

> *The postage evidencing device that performs printing the evidence of postage need not have stored therein or have access to secret keys, with the exception, if desired, of session keys needed for accessing information from protected tokens storage.*

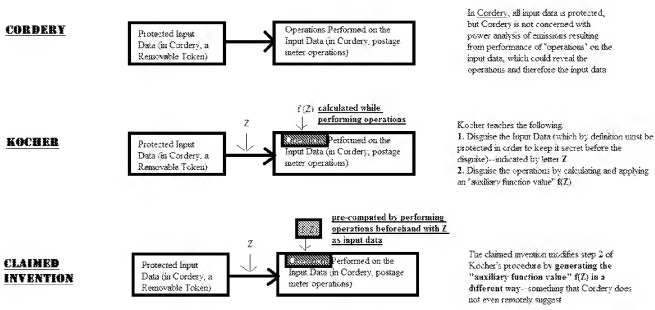
Rather than disguising data and operations, as in Kocher, Cordery teaches the generation of digital tokens that permit the printing information to be stored on a removable or portable (and therefore protectable) device 104 that needs to be inserted into the postage machine 102 during postage printing operations. In other words, Cordery is concerned with keeping secret input data secret by physically protected the data. **Cordery actually teachings away from performing operations with secret data in the open**. ***On the other hand, Kocher is specifically concerned with performing operations in the open, which is why Kocher provides ways to disguise the data used in the operations, and the operations themselves***.

The data protected by Cordery may be thought of as input data for the postage printing operations, and it does not correspond to the auxiliary function values of Kocher, which are used to disguise the operations **as they are being performed**. Based on the teachings of Kocher, even if Cordery were to attempt to disguise the postage printing operations performed by machine 102, there is no reason why Cordery would modify the algorithm of Kocher by changing the way in which auxiliary function values are calculated. The auxiliary function values are not the input data that needs to be kept secret. While it would not be contrary to the teachings of Kocher to store input data on a removable token as taught by Cordery, the input data that would be stored would be the input data subject to disguise by value Z, and not the auxiliary function values f(Z), which Kocher generates during operations and therefore sees no need to disguise. Cordery's teachings have no applicability to the calculation of auxiliary function values for the purpose of disguising the operations being performed on already-protected input data.

The Cordery patent appears to be confused about Appellant's arguments as to why the teachings of Cordery cannot be combined with those Kocher. This is not a matter of "non-analogous art" simply because Cordery is in the field of postage metering. Appellant has never argued that. The reason that Cordery does not render the claimed invention obvious is because the data being protected, namely input data for an operation, is not the data with which the claimed invention is concerned, namely auxiliary function values that disguise operations that use protected input data. **Kocher protects input data, just like Cordery does, but does not protect auxiliary function values.** Cordery, on the other hand, does not recognize, or have anything to do with, the problem of protecting auxiliary function values of the type taught by Kocher. Instead, Cordery merely teaches that certain digital tokens with protected data should be generated in a secure environment and stored on a portable device.

That input data should be protected before being subject to disguising and input to disguised operations as taught by Kocher would certainly not have been a surprise to an ordinary artisan reading Kocher. It is not even necessary to turn to Cordery for a teaching that secret data should be kept secret. However, Kocher goes way beyond protection of input data ***before*** it is used in an operation (whether a smartcard operation as in Kocher or a postage operation as in Cordery), by disguising the operation **as it is being performed**, ***after*** it is has left the secure environment. Cordery disguises no operations, and only protects the input data to the operations before being used in the postage printing machine. Because Cordery takes the approach of never performing certain operations in an insecure environment, Cordery is not at all concerned that the otherwise protected input data might be discovered by analyzing power emissions from the postage meter. Even if Cordery were concerned with this problem, Kocher offers a ready made solution–use of auxiliary function values to disguise the operations being performed on the input data provided upon insertion of the removable token into the postage meter. There is **no apparent need**, in any of the references of record, to modify the manner in which Kocher generate the auxiliary function values used to disguise operations *as they are being performed* on the secret and disguised input data.

These differences between the teachings of Cordery and Kocher, and the claimed invention, may perhaps be better understood from the following illustration:



The differences do not lie in the fact that Cordery concerns a postage meter. Postage meters perform operations using secret data. Further, the combination of Kocher and Cordery could be taken to suggest that Cordery's operations could be disguised using the auxiliary function value procedure taught by Kocher, or that Kocher's procedure could be applied to secret data stored on a portable or removable token as taught by Cordery. However, in either case, the claimed invention would not have resulted because Cordery does not suggest modification of Kocher's auxiliary function value generation procedure, but only concerns the data input to the operations that are being disguised..

The point of Kocher is to protect operations as they are being performed. This is fundamental to the teachings of Kocher, since the sole purpose of the auxiliary function values is to disguise the operations so that they are not revealed by power analysis that can only be carried out while the operations are being performed. No teaching in Cordery can change that fundamental purpose of Kocher. Cordery can only teach a way of protecting input data before the operations of

Kocher are carried out, *i.e.* before disguising the input data and the operations themselves. If the operations themselves were pre-performed, there would be no need for auxiliary function values in the first place. Cordery therefore could not possibly have suggested pre-performing the disguising operations. If Cordery does not suggest pre-performing the disguising operations, then why would it suggest pre-performing just a part of the disguising operations of Kocher, namely the generation of auxiliary function values to disguise the operations? Such a piecemeal modification of Kocher's teachings makes no sense.

In response to similar arguments made in previous responses, the Examiner simply relies on "**broader teachings**." This is explained in the paragraph bridging pages 3-4 of the Official Action, in which the Examiner responds to Appellant's arguments concerning the specific teachings of Cordery by pointing out that the Official Action in fact did not consider the specific teachings of Cordery. As explained in the Official Action,

> *It is noted that the previous Office action did not provide the specific analysis/conclusion that Appellant alleges, but rather relied upon the broader teaching in Cordery of secret function values being pre-computed (as previous cited, see Cordery, column 3, lines 18-25, where tokens are pre-computed, see also column 5, lines 10-12, where tokens include encrypted data, and column 3, lines 11-13, where the encryption algorithm and keys are protected.*

The Examiner goes on to respond to the Appellant's argument that there is no motivation for combining what is actually taught in Kocher (a power-analysis-preventing algorithm, which has a key storage arrangement) with what is actually taught in Cordery (a removable device for storing tokens and performing printing operations in a postal meter) by stating that he does not understand the arguments concerning what is actually taught by the references, implying that the teachings of the specific references are **irrelevant** to whether it would be obvious to combine them. In addition, the Examiner repeatedly points out that references from different fields can be combined if the problems are reasonably pertinent, and that it therefore does not matter that Kocher concerns protection of operations on a chip by power analysis and Cordery does not. **In reply, the Appellant respectfully submits that the Examiner is taking an entirely incorrect wrong approach to the obviousness analysis**. It DOES matter what is taught by the references.

As explained in MPEP 2141.02, p. 2100-107, "[a] prior art reference must be considered in its entirety, i.e., as a whole, including portions that would lead away from the claimed invention" (emphasis in the original). As explained in the previous response, when considering the teachings of Kocher as a whole, it can be seen that the array *dataIn* described in paragraphs [0068] to [0073] corresponds to the input data, the random bits *b* correspond to the auxiliary data, and the permutations defined by the arrays *table* and *perm* arguably correspond to the one or more operations of amended claim 26. The array *perm* represents an additional permutation that is computed randomly while computing the actual permutation defined by *table*. The output data *dataOut* representing a permutation of *dataIn* according to *table* are in fact not affect by *perm* although *perm* is twice applied to the input data. As described in paragraphs [0067], [0068], the additional permutation *perm* is used to avoid processing the steps to compute the permutation *dataOut* in input order or in output order since both orders may lead to leakage of information, so that *dataOut* itself does not depend on the random array *perm*, but rather it is the order in which *dataOut*'s entries are computed that depends on the random array *perm*. The falsification of data, on the other hand, is described by adding a random bit *b* modulo 2 to the permuted input data *perm [i]*, and storing the falsified bit in an array *temp*. This is expressed in Kocher as *dataIn [p] ^ b = dataIn [perm[i]] ^ b (*where ^ is the modulo operation). Thus, the step of falsifying data in Kocher, *i.e.*, blinding a bit of the input data by adding a random bit b modulo 2 (as can be seen in the third for-loop of the pseudo-code in paragraph [0068]), **is only performed AFTER the step of performing the additional permutation *perm***, meaning that the corresponding auxiliary data and function value computation steps cannot be carried out before at least one of the operation steps (obtaining *perm*) is performed. Thus, an **essential** feature of the method of Kocher in order to prevent information leakage (paragraphs [0068] and [0069] and cannot be omitted without rendering the method of Kocher inoperative. **This is what is actually taught by Kocher**. On the other hand, Cordery does not contain a single teaching that has anything specific to do with data corresponding to dataIn or dataOut of Kocher. Cordery merely teaches that keys and other secret data can be stored, which Kocher already does in memory 290, but cannot be reasonably said to suggest

changing the order of the third FOR loop in paragraph [0068] of Kocher and the step of performing the additional permutation *perm*.

Nothing in the Cordery patent would have caused the ordinary artisan to ignore the teaching in Kocher that an appropriate unblinding vector is stored in the array *dataOut* and already computed together with the blinded input vector, *i.e.*, in the same for-loop defined by *dataOut ptable[p] := b*. To the contrary, the Examiner's conclusion that one of ordinary skill in the art would have been caused by Cordery to ignore these teachings is pure conjecture. The Examiner will note that the left hand side equals *dataOut [table[perm[i]]*, *i.e.*, that the unblinding vector is determined by applying the permutations *perm* and *table* to the random vector *b* (the random bit *b* in step *i* of the for-loop being interpreted as an entry *b[i]* of a respective vector *b*). After the permutation defined by the array *table* is applied as the second of the one or more operations, to the falsified input data in the fourth for-loop, the appropriate compensation for the prior falsification of the input data follows by means of the auxiliary function value, namely the already computed value that was stored in the array entry *dataOut [table [p]]* in the previous loop as described above. Prior to these steps, the permutation array *perm* is once more randomly permuted (in the last for-loop on page 7). This procedure ensures that the order in which the steps in the following loop are executed again is different from the previous order of steps. However, such as an arrangement is optional and only serves to further avoid information leakage. The value of *dataOut* is not affected by this procedural step.

Thus, based on the actual teachings of Kocher, it must be concluded that Kocher specifically requires calculation of the auxiliary function value during data falsification, and that the auxiliary function values cannot be pre-stored and retrieved from a memory in the manner claimed. In particular, based on the actual teachings of Kocher, it can be seen that in order to modify the method of Kocher to obtain the claimed invention, a number of changes would have needed to be made, none of which are suggested by Cordery:

a. the ordinary artisan would have had to recognize that the method of Kocher may, at least in principle and despite explicit teachings to the contrary, be changed without any loss of security and without changing the output values by pre-computing the random bits *b and* the random permutation *perm* so that *b* and *perm* would serve as input data in addition to the actual input data *dataIn*, *dataOut*, and *table* (which would require considerable algorithmic skills not even remotely taught by Cordery);

b. the ordinary artisan would have had to further recognize that the blinding bits *b* would need to be pre-computed in safe surroundings and stored in an array of random blinding bits, for simplicity also called *b*, and that the random permutation *perm* would also have to be pre-computed in safe surroundings;

c. the ordinary artisan would have had to further recognize that the unblinding vector, stored in the vector *dataOut*, would have had to be pre-computed by applying the permutation *perm* and the permutation *table*, in that order, to the random vector *b*, *i.e.*, *dataOut [table [perm[i]] := b[i]*; and

d. the ordinary artisan would have had to provide for storing the random vector *b* representing the auxiliary data along with the unblinding vector *dataOut* representing the auxiliary function value, with the result that the main routine to compute the actual permutation of the input array *dataIn* according to the array *table* would then comprise the following blinding steps:

```
for (i=1; i<64; i++){
p=perm[i];                    //perm has already been pre-computed
temp [p] := dataIn[p] ^ b[i];        //random vector b[i] has
                                     // already been pre-computed}
```

Furthermore, even if the main routine of Kocher were modified in such a manner, and there is nothing in Cordery to suggest such a modification, the result would still not have been the claimed invention because the blinding step occurs only *after* the permutation *perm*, representing one of the one or more operations, has been applied to the input data *dataIn.* This wold be contrary to the teachings of Kocher since the execution of the permutation *perm* before blinding serves the security

purpose of randomizing the order in which the blinding steps are performed. On the other hand, the pre-computed unblinding vector would then simply read *dataOut[table[i] :=b[i]* since the respective application of the permutation *perm* would also have to be omitted in order to ensure a correct un-blinding step, resulting in a contradiction that renders the propose modification of Kocher inoperative. This conclusion follows logically from the teachings of Kocher.

The Appellant is thus not arguing that Cordery would not have suggested modification of Kocher simply because it is from a different field, but rather that none of the teachings in Cordery would have led the ordinary artisan to modify Kocher's power analysis protection algorithm. Kocher is not unaware that keys need to be kept secret. Kocher in fact uses secret keys. What Kocher does not appreciate is that contemporaneously computing auxiliary function values leaves chip operations, and therefore the secret data input to those operations, vulnerable to detection based on an analysis of radiation from the chip. Where in Cordery is there any teaching that would cause the ordinary artisan to realize that the contemporaneous calculation of auxiliary function values used to disguise operations performed in an insecure environment leaves a chip vulnerable? **There is no such teaching.** Kocher, on its face, seems to provide a perfectly good way of protecting data, and Cordery contains no teaching that would have suggested otherwise. Cordery's pre-storage of a token is not accomplished for the purpose of protecting a chip from "power analysis" by analyzing radiation emitted from the chip. Kocher, on the other hand, clearly recognizes that keys (like the ones in Cordery) should be kept secret. If Kocher's vulnerability had to do with protecting secret keys, *i.e.*, if Kocher needed a way to prevent someone from gaining access to the keys by reading them from memory, then application of Cordery's teachings of storing the keys on a removable token would make sense. However, the need to perform all operations in a secure environment is not the problem in Kocher. To the contrary, Kocher is concerned with the problem of how to perform the problems in an insecure environment (any public smartcard reader).

In other words, the Appellant is not arguing that Cordery is irrelevant because it concerns a postage meter, or storage of secret keys on a portable device. The Appellant is arguing that what

is taught by Cordery, namely storage of secret keys on a portable device, no matter how "generally" taken as argued by the Examiner, would not have caused the ordinary artisan to modify Kocher's algorithm for protecting data and operations from being discovered through external power analysis by pre-calculating the auxiliary function values used to protect the data rather than calculating the values as the operations are being performed. **The power analysis problem addressed by Kocher is not a problem of protecting secret keys *before* performing an operation (which is the problem that Cordery's removable token addresses), or of protecting the keys after the operation is performed, but rather of protecting the data during performance of the operation from eavesdroppers who analyze the radiation patterns emitted by the chip during performance of the operations. If the operations are repeated, then the eavesdropper can, by statistical analysis, figure out what operations are being performed. Once the operations are known, then the eavesdropper can determine which components of the radiation pattern involves the operations and which is the result of secret data/keys, and therefore discovery the secret keys or data. Merely performing operations, or storing secret data, on a protected device, as taught by Cordery, will not solve this problem.** On the other hand, Kocher does offer a solution to the problem–namely the use of auxiliary function values. These auxiliary function values are used to disguise operations that are being performed, and therefore Kocher generates them as needed for the operations. Nothing in Cordery says that this is a problem. Cordery concerns storage of secret data that might be used in operations and not detecting the operations themselves and working backward to discover the auxiliary function data used to protect the operations.

The issue that must be determined in this case is not, as alleged by the Examiner, whether Cordery has anything to do with protection of secret data (it does) or whether Cordery's data can actually be used in Kocher's device (it clearly cannot, but bodily incorporation is not the test for obviousness), but rather whether the teachings of Cordery, considered as a whole, would have led one of ordinary skill in the art to the claimed invention. In this case, the issue is:

- **whether Cordery' teaching of performing operations and protecting secret data through the use of removable storage devices would have led one of ordinary skill in**

the art to modify Kocher's method of protecting data during operations in an insecure environment by replacing contemporaneous auxiliary function value generation with predetermined auxiliary function values and retrieval of the auxiliary values from a memory before beginning data falsification?

Based on the above consideration of the actual teachings of Kocher, this issue must be answered in the negative. Cordery does not teach that there is a risk to generating auxiliary function values for use in disguising data operations performed by a processor. The processor of Cordery is a secure processor. All that Cordery teaches is a particularly secure way of storing predetermined sensitive data, by using an SPSD. Kocher also uses predetermined sensitive data, but how that predetermined sensitive data is determined has no effect on disguising of processor operations. In fact, Kocher is not concerned with the protection of pre-stored data. Prestored data cannot be compromised by power data analysis. Only actual processing operations generate power emissions that can be analyzed to discover secret data.

Both Kocher and the present invention are concerned with discovery of data protection algorithms, and the secret data used therein, by analyzing power emissions resulting from operation of a processor in an insecure environment. Kocher teaches one way to do so, namely by generating auxiliary function values during data falsification, and using the auxiliary function values to disguise the operations being performed as well as the secret data used in the operations. The claimed invention proposed a different way to disguise the data operations, by predetermining the auxiliary function values. This solves a problem that Kocher clearly did not consider, namely that the same analysis methods used to compromise the operations might, in a more sensitive form, be used to discover the auxiliary values generated as part of the operations. Pre-stored auxiliary function values cannot be analyzed in this manner. The mere fact that pre-determination and storage of secret values is known, which is all that Cordery brings to the table, does not suggest modification of Kocher's method by replacing the auxiliary function value generation step with retrieval of the auxiliary function values from a memory where the auxiliary function values are stored with values that are used to falsify the data. Kocher already knows that secret data needs to be kept secret and

that predetermined data needs to be stored (which is why memory 290 is provide to hold the "key"). The fact that Cordery stores the values on a removable token, and uses a secure processor to process the values, is not logically suggestive of modifying the processing algorithm of Kocher to eliminate auxiliary value generation. What possible advantage could such elimination have, other than protection against discovery by statistical analysis, which is not recognized as a problem in Kocher (with respect to the auxiliary function values), and which is clearly not a problem in the secure system of Cordery.

Cordery teaches nothing more than provision of secure data on an SPSD 104, and carrying out of processing operations using the secure data in a "secure co-processor." The operations are not carried out in an insecure environment as in Kocher, and there is no need for auxiliary functions values. If there were a need for auxiliary function values, none of the teachings of Cordery would prevent the ordinary artisan from generating the auxiliary function values during data falsification rather than pre-computing the auxiliary function values. In other words, Cordery teaches storage of sensitive pre-determined data on a removable token. It does not teach that a particular item of sensitive data required by Kocher to be generated during a specific processing operation should be replace by pre-stored data.

The failure of Cordery to suggest any sort of operation-falsification auxiliary function value storage logically follows from the fact that Cordery does not teach, or in way require, any sort of processor operation-falsification. In contrast to the setting of the present invention (and of Kocher) in which the secret data to be protected by falsifying some input data are stored on a data carrier, Cordery's secret data to be protected (in the form of decryption algorithms and keys), ***is neither stored nor executed on the data carrier 104 of Cordery but rather on a secure co-processor 502 separate from the data carrier 104, and which is further protected by a tamper resistant housing 513*** (see col. 9, line 66 to col. 10, line 61 and Fig. 5 of Cordery). In other words, Cordery teaches protection of secret data by placing it in a separate tamper resistant, secure co-processor, which is completely contrary to the present invention, in which secret data on a chip is protected by falsifying

operations on the chip and not by adding an additional secure, tamper resistant chip. A major purpose of the method of Kocher, which is part of the teachings of Kocher "as a whole," is to perform computations in an insecure environment, without the need to perform the operations in a secure environment or, by implication, to add a secure co-processor such as the one provided by Cordery. The method of Kocher is different than that of the claimed invention in that auxiliary function values are generated as the operations are being disguised, and in fact specifically teaches, as explained above, that auxiliary function values should not be pre-stored. Since Cordery only teaches pre-computation in the general context of a secure co-processor and without pre-calculation of auxiliary function compensating data, Cordery could not have suggested any modifications of Kocher that would overcome the differences between the operation-disguising auxiliary-function-generating procedures of Kocher and the claimed invention, and reversal of the rejection of claims 26-33 and 42 is accordingly requested.

Conclusion

For all of the foregoing reasons, Appellants respectfully submit that the Examiner's final rejection of claims 26-33 and 42 under 35 U.S.C. §103(a) is improper and should be reversed by this Honorable Board.

Respectfully submitted,

BACON & THOMAS, PLLC

/Benjamin E. Urcia/

Date: April 19, 2011 By: BENJAMIN E. URCIA
Registration No. 33,805

BACON & THOMAS
625 Slaters Lane, 4th Floor
Alexandria, Virginia 22314

Telephone: (703) 683-0500

S:\Producer\beu\Pending Q...Z\V\VATER 700656\SecondAppealBrief.wpd

VIII. APPENDIX OF CLAIMS ON APPEAL

26. A method for protecting secret data stored in a memory of a semiconductor chip of a data carrier, said secret data serving as input data for one or more operations executed on the semiconductor chip, the execution of the one or more operations causing signals detectable from outside of the data carrier, the signals being dependent on the one or more operations and on the input data for the one or more operations, said method comprising the steps of:

- falsifying the input data by combination with auxiliary data (Z) before execution of the one or more operations (f) on the semiconductor chip,
- executing said one or more operations (f) on the semiconductor chip,
- retrieving an auxiliary function value ($f(Z)$) from said memory of said semiconductor chip of the data carrier,
- combining the output data determined by said executing of the one or more operations (f) with ~~an~~ said auxiliary function value ($f(Z)$) in order to compensate for the falsification of the input data,
- wherein the auxiliary function value ($f(Z)$) was previously determined by execution of the one or more operations (f) with the auxiliary data (Z) as input data in safe surroundings and stored along with the auxiliary data (Z) in the memory of the semiconductor chip of the data carrier.

27. A method according to claim 26, wherein the combination with the auxiliary function values ($f(Z)$) for compensating the falsification is performed at the latest directly before execution of an operation (g) which is nonlinear with respect to the combination generating the falsification.

28. A method according to claim 26, wherein the auxiliary data (Z) are varied, the corresponding function values being stored in ~~the~~ a memory of ~~the~~ a data carrier.

29. A method according to claim 28, wherein new auxiliary values (Z) and new auxiliary function values ($f(Z)$) are generated by combining two or more existing auxiliary data (Z) and auxiliary function values ($f(Z)$).

30. A method according to claim 29, wherein the two or more existing auxiliary data (Z) and auxiliary function values ($f(Z)$) that are combined to generate the new auxiliary values (Z) and new auxiliary function values ($f(Z)$) are each selected randomly.

31. A method according to claim 26, wherein pairs of auxiliary data (Z) and auxiliary function values ($f(Z)$) are generated by a generator without the operation ($f(Z)$) being applied to the auxiliary data (Z).

32. A method according to claim 26, wherein the auxiliary data (Z) are a random number.

33. A method according to claim 26, wherein the output data and the auxiliary function value are combined by an XOR operation.

42. A method according to claim 26, wherein the security-relevant operations are key permutations or permutations of other secret data.

IX. EVIDENCE APPENDIX

No evidence is submitted herewith.

X. RELATED PROCEEDINGS APPENDIX

No related proceedings have occurred, and none are pending.